Demand Forecasting Intermittent and Lumpy Time Series: Comparing Statistical, Machine Learning and Deep Learning Methods

Daniel Kiefer
ESB Business School, Reutlingen University
Daniel.Kiefer@Reutlingen-University.de

Florian Grimm
ESB Business School, Reutlingen University
Florian.Grimm@Reutlingen-University.de

Markus Bauer
Karlsruhe Institute of Technology (KIT)
Markus.Bauer3@Partner.Kit.edu

Clemens van Dinther
ESB Business School, Reutlingen University
Clemens.van_Dinther@Reutlingen-University.de

Abstract

Forecasting intermittent and lumpy demand is challenging. Demand occurs only sporadically and, when it does, it can vary considerably. Forecast errors are costly, resulting in obsolescent stock or unmet demand. Methods from statistics, machine learning and deep learning have been used to predict such demand patterns. Traditional accuracy metrics are often employed to evaluate the forecasts, however these come with major drawbacks such as not taking horizontal and vertical shifts over the forecasting horizon into account, or indeed stock-keeping or opportunity costs. This results in a disadvantageous selection of methods in the context of intermittent and lumpy demand forecasts. In our study, we compare methods from statistics, machine learning and deep learning by applying a novel metric called Stock-keeping-oriented Prediction Error Costs (SPEC), which overcomes the drawbacks associated with traditional metrics. Taking the SPEC metric into account, the Croston algorithm achieves the best result, just ahead of a Long Short-Term Memory Neural Network.

1. Introduction

Demand forecasts are essential for most companies, indeed effective forecasts can represent a competitive advantage in decision support, as these forecasts are used as an input for production, transportation, sourcing, and inventory planning as well as strategic purposes such as supply chain planning [1].

A demand forecast is the best estimate of a future demand for a defined period [2]. Any error in forecasting can be particularly harmful to companies, hence forecasts must be as precise as possible [1, 3]. If forecasts are considerably higher than the actual demand, the company will produce or stock too many products that cannot be sold, which leads to increased costs and tied-up capital. In turn, forecasts lower than the actual demand lead to a loss of business opportunities due to a lower service level resulting from longer lead times [1].

Intermittent time series are characterized by multiple non-demand intervals. Demand occurs sporadically but in more or less equal amounts [4, 5]. Major differences in the size of the actual demand are related to lumpy time series [6, 7]. Figure 1 illustrates these demand patterns. Such demand patterns are especially difficult to forecast [8]. However, they are very common in real business, for example in heavy machinery, respective spare parts, aircraft service parts, electronics, maritime spare parts [9], automotive spare parts [10] as well as (fashion) retailing [11].

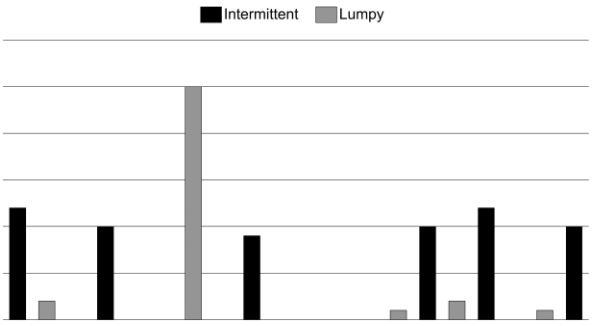

Figure 1. Intermittent and lumpy demand pattern [7]

In order to predict intermittent and lumpy time series, different approaches have been used with varying degrees of success. These include statistical methods such as Holt-Winters [12], or machine learning methods such as Support Vector Regression (SVR) [13] as well as deep learning methods such as Long Short-Term Memory (LSTM) Neural Networks [14]. However, it remains unclear which meta-level method or specific model is most suitable for forecasting intermittent and

URI: https://hdl.handle.net/10125/70784
978-0-9981331-4-0

lumpy time series. This is because intermittent and lumpy time series have not yet been sufficiently researched [15] and also as a result of the historical lack of appropriate metrics, which were deliberately developed for assessing demand forecasts of this time series pattern [7].However, in saying this, the research field of machine learning and deep learning has also evolved rapidly.

In this work, we apply methods from statistics, machine learning and the latest deep learning techniques to forecast demand of intermittent and lumpy time series and we analyze the advantages and disadvantages of each method in accordance with the Design Science Research (DSR) [16].

In addition, we assess the forecasts with a novel metric, the Stock-keeping-oriented Prediction Error Costs (SPEC), developed by Martin et al. [7] to evaluate demand forecasts of intermittent and lumpy time series. Evaluation using other metrics, for example the Mean Absolute Error (MAE), can lead to a disadvantageous selection of models, primarily because these do not account for (i) horizontal and vertical shifts in predictions over the forecasting horizon, (ii) temporal interaction between predictions of different points in time, and indeed (iii) opportunity or stock-keeping costs as regards units outstanding or in stock [7]. Moreover, we calculate the intermittence and lumpiness ratio of time series to gain a deeper understanding of which characteristics or magnitude of intermittence and lumpiness some methods perform better than others. For this purpose, a real-world data set is used.

The remainder of this work is organized as follows. In Chapter 2, we review the existing literature on forecasting methods, evaluation metrics and measurement methods for intermittent and lumpy time series. Furthermore, the existing research gap is highlighted. Chapter 3 addresses the experimental design of this article to deliver answers to the identified research gap. In Chapter 4, the results of the statistical, machine learning and deep learning prediction methods are analyzed using a real-world data set. Special attention is devoted to the novel SPEC metric and the measured degree of intermittency and lumpiness of the time series. Finally, we provide a conclusion in Chapter 5.

2. Related Work and Research Gap

For the analysis and preparation of the related work, the guidelines of Levy and Ellis [17] as well as Webster and Watson [18] are followed. Thus far, many different methods from statistics, machine learning and deep learning have been used to forecast intermittent and lumpy time series. In the following, promising models to forecast demand of intermittent and lumpy time series within the method categories are discussed in more detail.

Croston [4] examined forecasting methods for intermittent time series and concluded that the exponential smoothing methods used thus far are not particularly well suited. Based on this finding, he developed his own method, which is now used as a baseline in numerous analyses. In their investigations, Syntetos and Boylan [19] came to the conclusion that Croston's proposed method is biased and they therefore developed a new method. Further adjustments have since been made to Croston's original algorithm [20]. Despite some criticism, empirical studies have shown that Croston's method is superior to conventional methods [19, 21]. Other statistical forecasting methods that should be mentioned here include Holt-Winters [22, 23], Theta [24] as well as Autoregressive Integrated Moving Average (ARIMA) [25]. In the M4 Competition, these three methods are used as benchmarks on account of their good performance in time series prediction [26].

A machine learning approach that is frequently used and delivers good results in time series forecasting is Support Vector Regression (SVR) [27]. Hua and Zhang [28] combined the method with a logistic regression approach in which SVR predicted the occurrences of non-zero demand of spare parts. A study by Sapankevych and Sankar [29] demonstrated that it outperformed traditional statistical methods as well as deep learning techniques such as Multi-Layer Perceptron (MLP). Another machine learning method which was shown to achieve good forecasting results for time series is the XGBoost, an eXtreme Gradient Boosting framework [30]. It was the best method for electricity consumption prediction in a study by Deng et al. [31]. A quite similar machine learning method, the Random Forest, has also been successfully applied to forecasting electricity load, and has outperformed traditional statistical methods [32].

Deep learning methods are already successfully used for predicting time series and they have been shown to outperform classic statistical methods as well as machine learning methods [3, 33–36]. LSTM Neural Networks represent a further development of Recurrent Neural Networks (RNN), and were used for inventory forecasting by Abbasimehr et al. [35]. The results of the study on Neural Networks for demand forecasting intermittent time series by Kourentzes [3] were ambiguous due to different evaluation metrics. According to classic evaluation metrics such as the MAE, by comparison, the Neural Network was evaluated to be worse than the forecast result of Croston. However, where the service level was included as an evaluation metric, the Neural Network performed considerably better. Therefore, not only accuracy

metrics but also inventory metrics for insightful findings of intermittent demand are recommended [3, 33].

The selection of the evaluation metric is essential for the assessment of the forecast. Depending on the selected metric, the forecast or, specifically, its evaluation, can vary considerably. Choosing the suitable metric is exceptionally difficult, however. For example, the common Mean Absolute Percentage Error (MAPE) produces infinite or undefined values when actual values are zero, which are an integral part of intermittent time series [37].

Evaluating demand forecasts with traditional accuracy metrics can also result in misleading findings [3]. Hence, a novel metric, the SPEC, has been developed especially for evaluating demand forecast of intermittent and lumpy time series [7]. It closes the existing gaps mentioned in Chapter 1 of the RMSE, MASE, sMAPE, MAPE, MSE, MAE and so on. Therefore, a novel investigation of forecast methods with this evaluation metric is necessary.

Furthermore, it is important to analyze under what intensity of intermittency and lumpiness the different methods achieve better or worse results. To date, no studies have shown at what level of intermittency or lumpiness different methods achieve the greatest accuracy. Syntetos and Boylan [38] as well as Kostenko et al. [39] and Williams [40] have all dealt with the classification of demand patterns, especially intermittent and lumpy demand patterns. For the purposes of directly calculating a ratio for each time series regarding intermittence and lumpiness, Williams [40] is most suitable.

Nikolopoulos [15] also highlights the existing research gap in the context of studying demand forecasting models in the field of intermittent and lumpy time series. At the same time, the methods in Information Systems are rapidly developing. Hence, new developments in the field of deep learning should also be considered. The identified research gap leads to the following research questions:

RQ 1: Do modern advanced deep learning methods achieve considerably better forecasts than classic, established statistical methods and machine learning methods in forecasting demand for intermittent and lumpy time series?

RQ 2: Under which time series characteristics, in particular the degree of intermittent behavior and lumpiness of the time series, do deep learning methods achieve superior results and vice versa?

We conduct an empirical study using a publicly available data set that includes intermittent and lumpy time series. Subsequently, promising forecasting methods are examined using a novel suitable evaluation metric. The time series are then divided into classes based on the degree of intermittence and lumpiness. Thus, it is not only possible to identify the best, overall method, but also to analyze which methods are specifically suitable for different magnitudes of intermittence and lumpiness.

3. Suggested Experimental Design

In order to answer the aforementioned research questions and expand upon existing investigations, we propose an experimental design that is adapted to the shortcomings mentioned in the previous chapters regarding an appropriate metric as well as the highlighted methods. Figure 2 illustrates the suggested experimental design.

For reasons of transparency, publicly available data is used. The M5 Competition[1] is particularly suitable for this purpose, as it primarily contains intermittent and lumpy time series. The data is provided by Walmart and it comprises around 100,000 hierarchical daily time series at the SKU level with a length of 1,941 time-steps for each series (even if external feature data is available, we only use the univariate time series in this experiment).

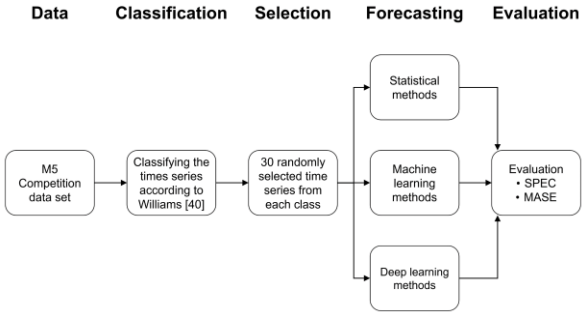

Figure 2. Suggested experimental design

These time series are classified using the approach described by Williams [40]. In order to calculate the intermittence degree of a time series, the following formula is used:

$$\frac{1}{\lambda \bar{L}} \tag{1}$$

with:

- λ mean (Poisson) demand arrival rate
- $\bar{L}$ the mean lead time duration

[1]The Makridakis Competitions are a series of open competitions organized by Spyros Makridakis to evaluate and compare the accuracy of different forecasting methods.

The higher the ratio, the more intermittent the demand. In order to calculate the lumpiness, the following formula is proposed:

$$\frac{CV_2(x)}{\lambda \bar{L}} \quad (2)$$

with:

- $CV_2(x)$ squared coefficient of variation of demand sizes

For the cut-off values, we adopt the values proposed by Williams, although we do not split Class D into D1 and D2 but rather retain one class (D). Figure 3 below illustrates the cut-off values and the resulting classes. Time series classified with an A show little intermittence and lumpiness, B show intermittence, while C have frequent demands of widely-varying sizes (lumpiness), and D are highly intermittent and lumpy [40].

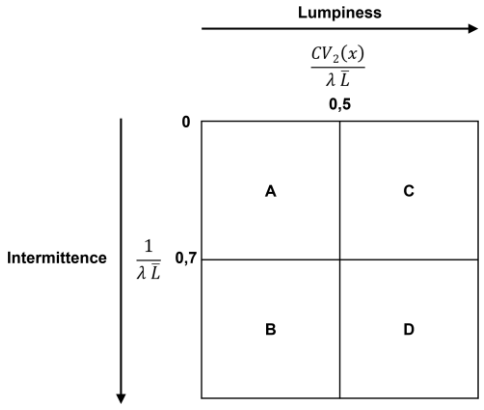

Figure 3. Categorization scheme [40]

Figure 4 displays the classification of all-time series in the M5 Competition. As expected, there are fewer time series in Class A because it is a data set specifically for intermittent and lumpy time series. If Class A is not considered, the other classes are relatively homogeneously distributed.

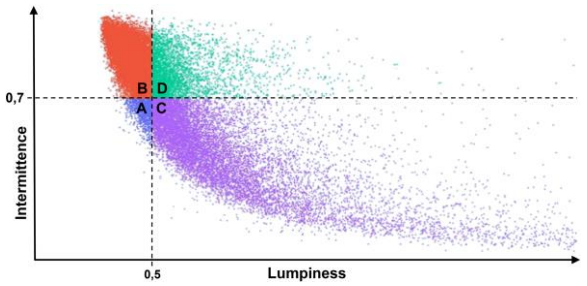

Figure 4. Intermittence and lumpiness classification

30 time series are randomly drawn from each class (in total 120 time series are considered) with the numpy function random.choice, which are then predicted by the methods presented below (before transferring the data as an input to the models they are scaled using the sklearn.StandardScaler).

As suggested by Bergmeir et al. [41], we evaluate the demand forecasts from the models on a rolling basis. In our experimental design, we operate on a four-fold basis. Figure 5 displays this approach.

The 30 randomly-chosen time series are predicted using statistical, machine learning and deep learning methods. It should be noted that the forecasts are made on a rolling basis as shown in Figure 5, in which the next 28 days are predicted. These forecasts are evaluated with the following metrics and the average value for the four-folds are calculated and discussed in Chapter 4.

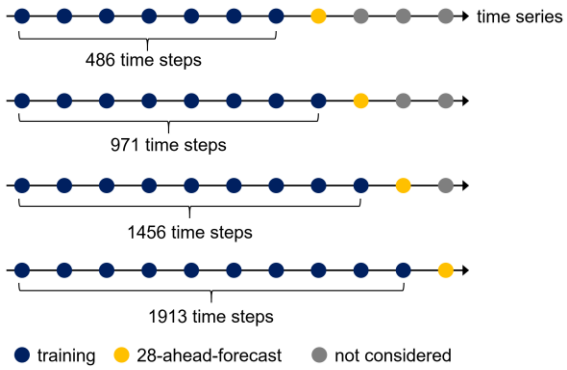

Figure 5. Forecasting on a rolling window

In Chapter 2, we highlighted the importance of a suitable metric for evaluating the prediction of intermittent and lumpy time series in demand forecasting. In this analysis we calculate two evaluation metrics: first, the SPEC [7], and second, the MASE [42]. The ranking for the evaluation of the forecasting methods is based on the SPEC, in line with the arguments made in Chapters 1 and 2.

$$SPEC_{a_1,a_2} = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=i}^{t}\left(max\left[0; min\left[y_i; \sum_{k=1}^{i} y_k - \sum_{j=1}^{t} f_j\right] \times a_1; min\left[f_i; \sum_{k=1}^{i} f_k - \sum_{j=1}^{t} y_j\right] \times a_2\right] \times (t-i+1)\right) \quad (3)$$

with:

- n length of time series
- y_t actual demand at time t
- f_t corresponding forecast
- $a_1 \epsilon [0, \infty]$ opportunity cost
- $a_2 \epsilon [0, \infty]$ stock-keeping cost

Martin et al. [7] recommend selecting a_1 and a_2 such that their sum is 1 (suggested relationship a_1 = 1- a_2). In the case study the authors conclude that $a_1 = 0.75$ and $a_2 = 0.25$ are effective parameters for the evaluation of demand forecast. For further explanation, we strongly recommend the article by Martin et al. [7].

The MASE [42] serves merely as a further comparison and basis for discussion.

$$MASE = mean(\left|\frac{e_t}{\frac{1}{n-1}\sum_{t=2}^{n}|Y_t - Y_{t-1}|}\right|) \quad (4)$$

with:

- Y_t observation at time t
- F_t forecast of Y_t
- e_t forecast error for the period ($e_t = Y_t - F_t$)
- $t = 1, \dots n$
- $F_t = Y_{i-1}$ (one-step naïve forecast method)

When MASE < 1, the proposed method results in smaller errors than the one-step naïve forecast method. Martin et al. [7] note that the interpretability of the MASE is difficult, especially in the context of demand forecasts of intermittent and lumpy time series. Through the comparison with the one-step naïve forecast, which will predict many zero values, the MASE value can be >1, although occurring demand was never correctly predicted by the naïve forecast, only the non-occurring demand (zero values).

The following paragraphs present the models from the methods statistics, machine learning and deep learning that are used for the proposed experiment. The models were selected based on their forecasting ability for intermittent and lumpy time series. These are widely used in the literature and also to answer the research questions in this article. The results of the models are presented in Chapter 4. Table 6 in Chapter 7 contains the selected parameters for the respective models to forecast demand (for parameters that are not listed, the default value is used).

Croston's [4] method is a statistical method developed to forecast demand of intermittent time series. Initially, the average size of demand is estimated using exponential smoothing. Next, the average interval between demands is calculated. This is then used in the form of a constant model to predict future demand. It should be pointed out that the Croston method does not forecast probable periods with non-zero receivables. This method assumes that all periods have demand with equal probability. It uses exponential smoothing to smooth the interval between demand and non-zero demand separately, but updates both only when there is non-zero demand. The in-sample adjustment and point forecast are then essentially the ratio of the smoothed non-zero demand divided by the time interval between the demands.

The Holt-Winters [22, 23] method is a triple exponential smoothing approach. Gamberini et al. [12] used it to forecast sporadic demand pattern successfully. In our experiment we use the model from the library statsmodels.tsa.holtwinter.

The ARIMA is a well-known forecasting method used both by scholars and in business applications. ARIMA models are linear, time-discrete models for stochastic processes. They are primarily used for statistical forecasting of time series, especially in economics, social sciences and engineering [25]. In our experiment we use the Auto-ARIMA model from the library pmdarima and the imported auto_arima.

XGBoosting [30] is a sequential technique that works on the principle of an ensemble. It combines a number of weak learners and offers improved forecasting accuracy. In our experiment we use the model from the library xgboost and the imported xgb.

Random Forest [43] is a classification method consisting of several uncorrelated decision trees. All decision trees have grown under a certain type of randomization during the learning process. For a classification, each tree in that forest is allowed to make a decision and the class with the most votes decides the final classification. Random Forests can also be used for regression; hence, it is possible to use the Random Forest for time series prediction. In our experiment we use the model from the library sklearn.ensemble and the imported Random-Forest-Regressor.

A Support Vector Machine (SVM) is mainly used as a classification method such as Random Forest but there is also the possibility for a regression, meaning both can be used for time series forecasting. Pai et al. [13] used and compared an SVM-Regression for forecasting seasonal time series. They concluded that an SVM is well-suited for this type of task. In our experiment we use the model from the library sklearn.svm and the imported SVR.

An MLP consists of more than one layer and neurons. The simple perceptron is a simplified artificial Neural Network first introduced in 1958 [44]. The basic version consists of a single artificial neuron with adjustable weightings and a threshold value. It converts an input vector into an output vector and thus represents a simple associative memory. In our experiment we use the model from the library sklearn.neural_network and the imported MLP-Regressor.

LSTM Neural Networks are often used to forecast time series. Due to their storage capacity and sequential cell operation, they are particularly suitable. They consist of one input, one forget, one remember gate as well as one output gate. In this way, in contrast to conventional Recurrent Neural Networks (RNNs), LSTMs enable a kind of memory of past experiences. Abbasimehr et al. [35] used LSTMs with great results in demand forecasting time series. In our experiment we use the model from the library tensorflow 2.0. Given the good prerequisites of LSTMs, two models are

developed and evaluated in Chapter 4. The second model receives an additional LSTM layer to analyze how and to what extent further layers and neurons can improve the prognosis.

4. Results

Table 1 below provides the ranks of the tested models evaluated with the SPEC for all forecasted time series (ranking is based on SPEC, see Chapter 3). Considering the novel SPEC (a lower value is better), it is clear that the statistical Croston algorithm performs best. Even the second-best model, the LSTM, has a 26% higher score.

Comparing the result with a classic evaluation metric, the MASE (a lower value is better), the LSTM has a 54% lower value and is therefore superior compared to Croston. At the same time, the MASE value from the LSTM is < 1 and thus better than the naïve forecast.

Table 1. Result of all classes

	Ø SPEC		Ø MASE		Rank
Statistic					
Croston	4.75	(0%)	2.15	(0%)	1
Holt-Winter	7.41	(56%)	1.08	(50%)	4
Auto-ARIMA	6.93	(46%)	1.04	(52%)	3
Machine Learning					
Random Forest	8.19	(73%)	1.14	(47%)	5
XGBoost	12.01	(153%)	1.13	(48%)	9
Auto-SVR	9.98	(110%)	1.10	(49%)	7
Deep Learning					
MLP	10.15	(114%)	1.66	(23%)	8
LSTM	5.96	(26%)	0.98	(54%)	2
LSTM-2	8.57	(81%)	1.04	(52%)	6

It is surprising that the LSTM-2 achieves worse results than the LSTM considering the SPEC. It seems that the additional layer with 28 neurons could not enhance the quality of the demand forecast in this setup. On the other hand, the LSTM-2 achieves the second best MASE value and is on par with the Auto-ARIMA model. Both achieved a 52% lower score than the Croston algorithm and are therefore better. However, regarding demand forecast the SPEC is more suitable for the selection of the best adequate model.

Taking this metric into account, the Auto-ARIMA comes in third place and the Holt-Winter algorithm in fourth place. The statistical methods thus dominate the upper ranks compared to the other methods.

The deep learning methods perform better on average with the SPEC metric than the machine learning methods. An exception is the Random Forest, which is in fifth place compared to all methods, thus better than the MLP as well as the LSTM-2.

Comparing the MASE values, all methods are relatively close together. Only the Croston algorithm and the MLP scored particularly low. The remaining models achieve a better value than Croston within the range of 47–54%. Considering the SPEC metric, a striking result is made by the statistical methods: overall, they rank first. Deep learning ranks second and, overall, machine learning methods rank third.

In the following, the individual forecast results of the four classes according to Williams are discussed. This should provide a better understanding of which models can handle which degree of intermittency and lumpiness, as well as how well the models are able to forecast them.

Table 2 presents the ranks for the tested models considering the SPEC of the classified time series A (low intermittence and low lumpiness). Croston as well as the LSTM model did not change the rank, but the percentage differences declined and the result of the LSTM is now even closer to the best model, Croston.

Croston's lead is smaller in Class A compared to the other classes. The Holt-Winter and Auto-ARIMA models switched ranks. Thus, Holt-Winters triple exponential smoothing approach worked better for time series which are not intermittent and not lumpy (Class A) compared to the Auto-ARIMA. In this time series Class A, the LSTM-2 does not profit from the additional LSTM layer.

Table 2. Class A

	Ø SPEC		Ø MASE		Rank
Statistic					
Croston	2.02	(0%)	1.85	(0%)	1
Holt-Winter	3.45	(71%)	1.06	(43%)	3
Auto-ARIMA	3.76	(86%)	1.03	(44%)	4
Machine Learning					
Random Forest	4.29	(112%)	1.04	(44%)	7
XGBoost	5.25	(160%)	1.16	(37%)	9
Auto-SVR	4.05	(100%)	1.00	(46%)	6
Deep Learning					
MLP	4.31	(113%)	1.54	(17%)	8
LSTM	2.24	(11%)	0.96	(48%)	2
LSTM-2	3.99	(97%)	1.03	(44%)	5

The results of Class B (intermittent) can be seen in Table 3 below.

Table 3. Class B

	Ø SPEC		Ø MASE		Rank
Statistic					
Croston	10.42	(0%)	1.59	(0%)	1
Holt-Winter	13.45	(29%)	1.16	(27%)	4
Auto-ARIMA	12.32	(18%)	1.12	(29%)	2
Machine Learning					
Random Forest	17.35	(67%)	1.27	(20%)	6
XGBoost	23.21	(123%)	1.20	(24%)	9
Auto-SVR	20.11	(93%)	1.09	(32%)	7
Deep Learning					
MLP	20.97	(101%)	1.49	(6%)	8
LSTM	12.38	(19%)	1.14	(28%)	3
LSTM-2	16.82	(61%)	1.07	(33%)	5

Regarding the SPEC values, Croston is again on the first rank. However, the Auto-ARIMA achieved a slightly better result than the LSTM with a 1% lower value. In this class, the Auto-ARIMA is also better than the LSTM considering MASE. The lowest MASE value was achieved by the LSTM-2, which ranks fifth for the more relevant SPEC.

Table 4 presents the results for Class C, which contains the lumpy time series. Croston's algorithm also achieves the best result measured by SPEC in this class. The LSTM network comes second with a 45% worse SPEC value. The Random Forest achieves third place with a 72% higher SPEC value compared to Croston. Clearly, Random Forest performs considerably better in this class than XGBoost and Auto-SVR. The other models achieve twice to three times worse results than Croston. If we consider the traditional MASE metric, the LSTM achieves the best (lowest) value and is the only model to achieve a value < 1. On average, the other models have a value about 50% better than Croston. It is noticeable here that the MLP performs particularly poorly.

Table 4. Class C

	Ø SPEC		Ø MASE		Rank
Statistic					
Croston	5.04	(0%)	2.18	(0%)	1
Holt-Winter	9.90	(96%)	1.07	(51%)	5
Auto-ARIMA	9.15	(81%)	1.04	(52%)	4
Machine Learning					
Random Forest	8.65	(72%)	1.15	(47%)	3
XGBoost	15.05	(198%)	1.10	(49%)	9
Auto-SVR	12.35	(145%)	1.06	(52%)	8
Deep Learning					
MLP	11.45	(127%)	1.68	(23%)	7
LSTM	7.29	(45%)	0.97	(56%)	2
LSTM-2	11.12	(121%)	1.04	(52%)	6

Table 5 provides the results for class D, which consists of intermittent and lumpy time series. Across all classes, Croston always achieves the lowest SPEC value compared to the other machine learning and deep learning models. With a 34% gap, the LSTM achieves second place. While the LSTM-2 did not perform particularly well in the other classes, it took third place in this class, with a 56% gap to Croston. Of the machine learning models, it is mainly the Random Forest that performs relatively well, ranking in fourth with a 64% worse score compared to Croston. The Auto-ARIMA model also performs relatively well in this class, with a 66% worse result than Croston. Yet it also achieves the lowest value, also < 1, in terms of the MASE metric. All other models achieve values >= 1, whereby both LSTM models stand out positively with the lowest MASE values after the Auto-ARIMA.

Table 5. Class D

	Ø SPEC		Ø MASE		Rank
Statistic					
Croston	1.51	(0%)	3.00	(0%)	1
Holt-Winter	2.83	(88%)	1.03	(66%)	6
Auto-ARIMA	2.49	(66%)	0.97	(68%)	5
Machine Learning					
Random Forest	2.46	(64%)	1.11	(63%)	4
XGBoost	4.54	(202%)	1.05	(65%)	9
Auto-SVR	3.40	(126%)	1.27	(58%)	7
Deep Learning					
MLP	3.87	(157%)	1.91	(36%)	8
LSTM	2.02	(34%)	1.00	(67%)	2
LSTM-2	2.35	(56%)	1.02	(66%)	3

Based on the results presented, it is evident that the Croston algorithm is well suited to demand forecasting of intermittent and lumpy time series. It has a considerably lower stock prediction error cost compared to the other models. Furthermore, the computational time is very low compared to the deep learning methods and the handling is simple, which also makes it well suited to demand forecasting.

5. Conclusion

According to the current state of research it is unclear which methods from statistics, machine learning and deep learning are well suited to predict the demand for intermittent and lumpy time series. Past research has mostly compared only a few methods, and traditional metrics have been used to evaluate the models. These are not suitable for this problem and lead to the disadvantageous selection of models. At the same time, it is important to understand the results of the models in relation to the degree of intermittency and the degree of lumpiness.

One main contribution of this work is the analysis of the performance of statistical, machine learning and deep learning methods to forecast intermittent and lumpy time series. To evaluate the performance a novel metric, the SPEC, developed for the purpose of evaluating demand forecasts of intermittent and lumpy time series was used. As a further basis for comparison, the MASE was calculated. In order to deliver more insights about the behavior of the methods, the time series were also classified by the level of intermittence and lumpiness. Therefore, it was possible to examine the results in more detail and make statements about the degree of intermittency and lumpiness, as well as which models perform better under which circumstances. The M5 Competition data set was used to provide empirical evidence of the performance from the assessed methods.

Referring to RQ 1 of Chapter 2, it can be argued that in our analysis, modern deep learning methods, and especially LSTMs, achieved good but not the best

results. The Croston algorithm achieved the best results considering the SPEC.

For RQ 2 in Chapter 2, the results of the established classes were considered in relation to the degree of intermittency and lumpiness. In the established Class D with intermittent as well as lumpy time series, deep learning procedures did not achieve directly superior results. RNN-specific LSTM architectures achieved second and third place but Croston's algorithm achieved the best results. In Class C with lumpy time series, Croston also placed first and LSTM architecture second while third place was taken by the Random Forest model with a 76% worse result compared to Croston. Class B with intermittent time series was again dominated by Croston. With a result 18% worse than Croston, the Auto-ARIMA method took second place here. The LSTM architecture was slightly worse with 19%. Across all of the time series, it is clear that the models from the statistical area achieved very good results. From the area of deep learning, the LSTM architecture is to be mentioned. The machine learning models achieved below average results and could not prevail over the statistical or deep learning models with the exception towards the MLP, which achieved poor results.

As far as MASE is concerned, the results differ from those of the SPEC metric, because in many cases machine and deep learning models achieve better results than statistical models. However, the traditional MASE metric has major drawbacks in selecting the best model in the context of demand forecasts in case of intermittent and lumpy time series (see Chapters 1 and 2).

By developing the SPEC metric, Martin et al. [7] have made an essential contribution, inspiring us to use it to perform further in-depth detailed analysis in the context of demand forecasting of intermittent and lumpy time series. The scholars' research focused on the newly developed metric itself, while in our case this metric is used for a new comprehensive comparison of methods to demand forecast intermittent and lumpy time series. Furthermore, the data set we used is publicly accessible and the examined methods as well as the parameters are also transparent. In addition, the classification of time series based on their degree of intermittency and lumpiness provides further important contributions to understand the suitability of a model.

The presented results of this holistic study help to better understand forecast methods in the context of demand forecasting intermittent and lumpy time series. Demand forecasting is highly relevant in the area of logistics and supply chain. Through the analysis of nine forecast models with the novel metric SPEC it could be shown that statistical forecast methods can achieve greater results than with the described machine learning and deep learning methods.

Our work provides new important insights, which are partly limited for various reasons and require further research. Due to the idea of transparency, a publicly accessible data set was used. It contains time series from Walmart. This data could contain a bias regarding the distribution or similar. Furthermore, it should be emphasized that univariate time series were used. By including additional external data, the results could lean in favor of machine learning and deep learning models. Although nine models from different methods for example machine learning have been tested extensively, they are not yet generally meaningful on a meta-level. Since rapid technological progress is being made, especially in the area of deep learning, and since there are also very successful hybrid models.

Further studies, particularly with hybrid models of deep learning methods like the winner [45] of the M4 Competition, should be conducted to explore and analyze more models to further develop them for intermittent and lumpy time series predictions.

6. References

[1] D. Ivanov, A. Tsipoulanidis, and J. Schönberger, *Global Supply Chain and Operations Management*. Cham, Germany: Springer, 2017.

[2] M.A. Moon, *Demand and supply integration: The key to world-class demand forecasting*. Boston, USA: De Gruyter, 2018.

[3] N. Kourentzes, "Intermittent demand forecasts with neural networks," *Int. J. Prod. Econ.*, vol. 143, no. 1, pp. 198–206, May. 2013.

[4] J.D. Croston, "Forecasting and Stock Control for Intermittent Demands," *Oper. Res. Quart. (1970-1977)*, vol. 23, no. 3, pp. 289–303, Sep. 1972.

[5] P. Wallström and A. Segerstedt, "Evaluation of forecasting error measurements and techniques for intermittent demand," *Int. J. Prod. Econ.*, vol. 128, no. 2, pp. 625–636, Dec. 2010.

[6] S. Mukhopadhyay, A.O. Solis, and R.S. Gutierrez, "The Accuracy of Non-traditional versus Traditional Methods of Forecasting Lumpy Demand," *J. Forecast*, vol. 31, no. 8, pp. 721–735, Aug. 2011.

[7] D. Martin, P. Spitzer, and N. Kühl, "A New Metric for Lumpy and Intermittent Demand Forecasts: Stock-keeping-oriented Prediction Error Costs," in *Proc. 53rd Hawaii Int. Conf. Syst. Sci.*, Maui, HI, USA, Jan. 2020, pp. 974–983.

[8] Q. Xu, N. Wang, and H. Shi, "Review of Croston's method for intermittent demand forecasting," in *9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, Sichuan, China, May. 2012, pp. 1456–1460.

[9] A.A. Syntetos and J.E. Boylan, "The accuracy of intermittent demand estimates," *Int. J. Forecast.*, vol. 21, no. 2, pp. 303–314, Jun. 2005.

[10] T.R. Willemain, C.N. Smart, and H.F. Schwarz, "A new approach to forecasting intermittent demand for service

parts inventories," *Int. J. Forecast.*, vol. 20, no. 3, pp. 375–387, Sep. 2004.
[11] C. Li and A. Lim, "A greedy aggregation–decomposition method for intermittent demand forecasting in fashion retailing," *Eur. J. Oper. Res.*, vol. 269, no. 3, pp. 860–869, Sep. 2018.
[12] R. Gamberini, F. Lolli, B. Rimini, F. Sgarbossa, and C. Cattani, "Forecasting of Sporadic Demand Patterns with Seasonality and Trend Components: An Empirical Comparison between Holt-Winters and (S)ARIMA Methods," *Math. Probl. Eng.*, vol. 2010, pp. 1–14, Jul. 2010.
[13] P.F. Pai, K.P. Lin, C.S. Lin, and P.T. Chang, "Time series forecasting by a seasonal support vector regression model," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4261–4265, Jun. 2010.
[14] W. Fu, C.F. Chien, and Z.H. Lin, "A Hybrid Forecasting Framework with Neural Network and Time-Series Method for Intermittent Demand in Semiconductor Supply Chain," in *Proc. Int. IFIP WG 5.7 Conf. Advances Prod. Manage. Syst.*, Aug. 2018, pp. 65–72.
[15] K. Nikolopoulos, "We need to talk about intermittent demand forecasting," *Eur. J. Oper. Res.*, Jan. 2020.
[16] V. Vaishnavi and W. Kuechler, *Design science research methods and patterns: Innovating information and communication technology,* 2nd ed. London, England: Taylor & Francis, 2015.
[17] Y. Levy and T.J. Ellis, "A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research," *Inf. Sci.*, vol. 9, pp. 181–212, 2006.
[18] J. Webster and R.T. Watson, "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quart.*, vol. 26, no. 2, pp. xiii–xxiii, Jun. 2002.
[19] A.A. Syntetos and J.E. Boylan, "On the bias of intermittent demand estimates," *Int. J. Prod. Econ.*, vol. 71, 1-3, pp. 457–466, May. 2001.
[20] E. Leven and A. Segerstedt, "Inventory control with a modified Croston procedure and Erlang distribution," *Int. J. Prod. Econ.*, vol. 90, no. 3, pp. 361–367, Aug. 2004.
[21] T.R. Willemain, C.N. Smart, J.H. Shockor, and P.A. DeSautels, "Forecasting intermittent demand in manufacturing: a comparative evaluation of Croston's method," *Int. J. Forecast.*, vol. 10, no. 4, pp. 529–538, Dec. 1994.
[22] P.R. Winters, "Forecasting Sales by Exponentially Weighted Moving Averages," *Manag. Sci.*, vol. 6, no. 3, pp. 324–342, Apr. 1960.
[23] C.C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. J. Forecast.*, vol. 20, no. 1, pp. 5–10, Mar. 2004.
[24] V. Assimakopoulos and K. Nikolopoulos, "The theta model: a decomposition approach to forecasting," *Int. J. Forecast.*, vol. 16, no. 4, pp. 521–530, Oct. 2000.
[25] M.Z. Babai, M.M. Ali, J.E. Boylan, and A.A. Syntetos, "Forecasting and inventory performance in a two-stage supply chain with ARIMA(0,1,1) demand: Theory and empirical analysis," *Int. J. Prod. Econ.*, vol. 143, no. 2, pp. 463–471, Jun. 2013.
[26] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods," *Int. J. Forecast.*, vol. 36, no. 1, pp. 54–74, Mar. 2020.
[27] J.P. Karmy and S. Maldonado, "Hierarchical time series forecasting via Support Vector Regression in the European Travel Retail Industry," *Expert Syst. Appl.*, vol. 137, pp. 59–73, Dec. 2019.
[28] Z. Hua and B. Zhang, "A hybrid support vector machines and logistic regression approach for forecasting intermittent demand of spare parts," *Appl. Math. Comput*, vol. 181, no. 2, pp. 1035–1048, Oct. 2006.
[29] N. Sapankevych and R. Sankar, "Time Series Prediction Using Support Vector Machines: A Survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, Apr. 2009.
[30] Y. Wang and Y. Guo, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," *China Commun.*, vol. 17, no. 3, pp. 205–221, Apr. 2020.
[31] W. Deng, G. Lyu, Y. Shi, and W. Wang, "Electricity Consumption Prediction Using XGBoost Based on Discrete Wavelet Transform," in *2nd Int. Conf. Artif. Intell. and Eng. Appl.*, Guilin, China, Sep. 2017, pp. 716–729.
[32] T. Ahmad and H. Chen, "Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems," *Sustain. Cities Soc.*, vol. 45, pp. 460–473, Feb. 2019.
[33] R.S. Gutierrez, A.O. Solis, and S. Mukhopadhyay, "Lumpy demand forecasting using neural networks," *Int. J. Prod. Econ.*, vol. 111, no. 2, pp. 409–420, Feb. 2008.
[34] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case," *arXiv*, pp. 1–10, Jan. 2020.
[35] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput. Ind. Eng.*, vol. 143, May. 2020.
[36] S.F. Crone, "Artificial Neural Networks for Time Series Prediction: A Novel Approach to Inventory Management Using Asymmetric Cost Functions," *IC-AI*.
[37] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *Int. J. Forecast.*, vol. 32, no. 3, pp. 669–679, Sep. 2016.
[38] A.A. Syntetos, J.E. Boylan, and J.D. Croston, "On the categorization of demand patterns," *J. Oper. Res. Soc.*, vol. 56, no. 5, pp. 495–503, May. 2005.
[39] A.V. Kostenko and R.J. Hyndman, "A note on the categorization of demand patterns," *J. Oper. Res. Soc.*, vol. 57, no. 10, pp. 1256–1257, Oct. 2006.
[40] T.M. Williams, "Stock Control with Sporadic and Slow-Moving Demand," *J. Oper. Res. Soc.*, vol. 35, no. 10, pp. 939–948, Oct. 1984.
[41] C. Bergmeir, R.J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Comput. Stat. Data Anal.*, vol. 120, pp. 70–83, Apr. 2018.
[42] R.J. Hyndman and A.B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[43] H. Altincöp and A. Oktay, "Air Pollution Forecasting with Random Forest Time Series Analysis," in *2018 Int. Conf. Artif. Intell. and Data Process.*, Malatya, Turkey, Sep. 2018, pp. 1–5.

[44] D. Aackley, G. Hinton, and T. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.
[45] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecast.*, vol. 36, no. 1, pp. 75–85, Mar. 2020.

7. Appendix

Table 6 below contains the selected parameters for the respective models to forecast demand (for parameters that are not listed, the default value is used).

Table 6. Selected parameters of the models

	Parameter	Value
Statistic		
Croston		
	a	0.4
Holt-Winter		
	Trend	Add
	Seasonal	Add
	Seasonal_periods	7
Auto-ARIMA		
	Start_p	1
	Start_q	1
	Max_p	3
	Max_q	3
	m	7
	Start_P	0
	d	1
	Seasonal	True
	D	1
	Trace	True
	Error_action	Ignore
	Suppress_warnings	True
	Stepwise	True
Machine Learning		
Random Forest		
	Bootstrap	True
	Criterion	MSE
	max_depth	50
	max_features	Auto
	max_leaf_nodes	None
	min_impurity_decrease	0.1
	min_impurity_split	None
	min_samples_leaf	10
	min_samples_split	2
	min_weight_fraction_leaf	0.0
	n_estimators	1000
	n_jobs	-1
	oob_score	False
	random_state	1
	verbose	False
	warm_start	False
XGBoost		
	n_estimators	1000
	Verbose	False
Auto-SVR		
	kernel	Rbf
	degree	3
	gamma	Scale
	Coef0	0.0
	tol	0.001
	C	1.0
	epsilon	0.1
	shrinking=True	True
	cache_size	200
	verbose	False
	max_iter	-1
Deep Learning		
MLP		
	hidden_layer_sizes	100,50,10
	activation	ReLu
	solver	Adam
	alpha	0.001
	batch_size	auto
	learning_rate	invscaling
	learning_rate_init	0.001
	power_t	0.5
	max_iter	1000
	shuffle	True
	random_state	1
	tol	0.001
	verbose	False
	warm_start	False
	momentum	0.9
	nesterovs_momentum	True
	early_stopping	False
	validation_fraction	0.1
	beta_1	0.9
	beta_2	0.999
	epsilon	1e-08
LSTM		
	Sequential_LSTM_Layer	28
	return_sequences	True
	Dense_Layer	1
	batch_size	64
	window_size	28
	epochs	50
	lr	0.1
	optimizers	SGD
	loss	Huber
LSTM-2		
	Sequential_LSTM_Layer	28
	return_sequences	True
	Sequential_LSTM_Layer	28
	Dense_Layer	1
	batch_size	64
	window_size	28
	epochs	50
	lr	0.1
	optimizers	SGD
	loss	Huber